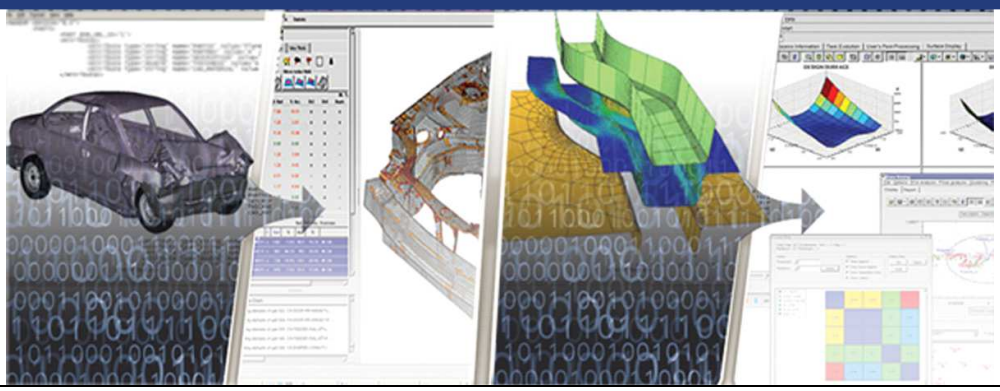


RADIOSS User's Code Interface

2017 version – January 2017

Appendix

Chapter 9

Altair Engineering, Inc., World Headquarters: 1820 E. Big Beaver Rd., Troy MI 48083-2031 USA
Phone: +1.248.614.2400 • Fax: +1.248.614.2411 • www.altair.com • info@altair.com

TABLE OF CONTENTS

9.1 – Appendix 1 – Available User's Arguments

9.1.1 Available Starter User's Arguments

Starter User's Arguments – Material Law for Shells				
Variable	***Size***	***Type***	***Write/read***	***Definition***
IIN	1	Integer	R	Input file unit (Starter input file)
IOUT	1	Integer	R	Output file unit (Starter output file)
UPARAM	NUPARAM	Float	W	User's parameter
MAXNUPARAM	1	Integer	R	Maximum size of UPARAM
NUPARAM	1	Integer	W	Size of UPARAM
NUVAR	1	Integer	W	Number of user's variables
IFUNC	NFUNCT	Integer	W	Function number array
MAXNFUNC	1	Integer	R	Maximum size of IFUNC
NFUNCT	1	Integer	W	Size of IFUNC
PARMAT	3	Float	W	(1) Stiffness modulus for interface (2) Young modulus (for shell) (3) Poisson's ratio (for shell)

Starter User's Arguments – Material Law for Bricks				
Variable	***Size***	***Type***	***Write/read***	***Definition***
IIN	1	Integer	R	Input file unit (Starter input file)
IOUT	1	Integer	R	Output file unit (Starter output file)
UPARAM	NUPARAM	Float	W	User's parameter
MAXNUPARAM	1	Integer	R	Maximum size of UPARAM
NUPARAM	1	Integer	W	Size of UPARAM
NUVAR	1	Integer	W	Number of user's variables
IFUNC	NFUNCT	Integer	W	Function number array
MAXNFUNC	1	Integer	R	Maximum size of IFUNC
NFUNCT	1	Integer	W	Size of IFUNC
STIFINT	1	Integer	W	Stiffness modulus for interface

Starter User's Arguments – Spring Property				
Variable	***Size***	***Type***	***Write/read***	***Definition***
IIN	1	Integer	R	Input file unit (Starter input file)
IOUT	1	Integer	R	Output file unit (Starter output file)
NUVAR	1	Integer	W	Number of user's variables
UVAR	NUVAR* NEL	Float	R/W	User's variable
PARGEO	*	Float	W	(1) Skew number (2) Stiffness for interface (3) Front wave option
NEL	1	Integer	R	Number of elements
IPROP	1	Integer	R	Property number
IX	3* NEL	Float	R	Spring connectivity
XL	NEL	Float	R	Element length

Starter User's Arguments – Spring Property				
Variable	***Size***	***Type***	***Write/read***	***Definition***
MASS	NEL	Float	W	Element mass
XINER	NEL	Float	W	Element inertia (spherical)
STIFM	NEL	Float	W	Element stiffness (time step)
STIFR	NEL	Float	W	Element rotation stiffness (time step)
VISCM	NEL	Float	W	Element viscosity (time step)
VISCR	NEL	Float	W	Element rotation viscosity (time step)

Starter User's Arguments – Solid Property				
Variable	***Size***	***Type***	***Write/read***	***Definition***
NEL	1	Integer	R	Number of elements
IIN	1	Integer	R	Input file unit (Starter input file)
IOUT	1	Integer	R	Output file unit (Starter output file)
NUVAR	1	Integer	W	Number of user's variables
UVAR	NUVAR* NEL	Float	R/W	User's variable
PARGEO	*	Float	W	(1) Skew number (2) Stiffness for interface (3) Front wave option
IPROP	1	Integer	R	Property number
IMAT	1	Integer	R	Material number
SOLID_ID	NEL	Integer	R	Solid Element id t=0
EINT	NEL	Float	W	Total internal energy t=0
VOL	NEL	Float	R/W	Initial volume
OFF	NEL	Float	R/W	Delete flag
RHO	NEL	Float	R/W	Density
SIG	6*NEL	Float	R/W	Stress tensor
XX1	NEL	Float	R	X coordinate node 1
YY1	NEL	Float	R	Y coordinate node 1
ZZ1	NEL	Float	R	Z coordinate node 1
XX2 . . ZZ8	NEL	Float	R	Same for node 2 to 8
VX1	NEL	Float	R	X velocity node 1
VY1	NEL	Float	R	Y velocity node 1
VZ1	NEL	Float	R	Z velocity node 1
VRX1	NEL	Float	R	X rotational velocity node 1
VRY1	NEL	Float	R	Y rotational velocity node 1
VRZ1	NEL	Float	R	Z rotational velocity node 1
MAS1	NEL	Float	W	Mass node 1
MAS2 . . MAS8	NEL	Float	W	Same for node 2 to 8
INN1	NEL	Float	W	Inertia node 1
INN2 . . INN8	NEL	Float	W	Same for node 2 to 8
STIFM	NEL	Float	W	Element stiffness (time step)
STIFR	NEL	Float	W	Element rotation stiffness (time step)
VISCM	NEL	Float	W	Element viscosity (time step)
VISCR	NEL	Float	W	Element rotation viscosity (time step)

Starter User's Arguments – Sensor				
Variable	***Size***	***Type***	***Write/read***	***Definition***
IIN	1	Integer	R	Input file unit (Starter input file)
IOUT	1	Integer	R	Output file unit (Starter output file)

9.1.2 Available Engine User's Arguments

Engine User's Arguments – Material Law for Shells				
Variable	***Size***	***Type***	***Write/read***	***Definition***
NEL	1	Integer	R	Number of elements
UVAR	NUVAR* NEL	Float	R/W	User's variable
UPARAM	NUPARAM	Float	W	User's parameter
NUPARAM	1	Integer	W	Size of UPARAM
NUVAR	1	Integer	W	Number of user's variables
NFUNCT	1	Integer	W	Size of IFUNC
IFUNC	NFUNCT	Integer	W	Function number array
NPF	*	Integer	R	Function array
NPT	1	Integer	R	Number of layers or integration points
IPT	1	Integer	R	Layer or integration point number
IFLAG	*	Integer	R	Geometrical flags
TF	*	Float	R	Function array
NGL	NEL	Integer	R	Element number
TIME	1	Float	R	Current time
TIMESTEP	1	Float	R	Current time step
RHO0	NEL	Float	R	Initial density
AREA	NEL	Float	R	Area
EINT	NEL	Float	R	Total internal energy
THKLY	NEL	Float	R	Layer thickness
EPSPXX	NEL	Float	W	Strain rate XX
DEPSXX	NEL	Float	W	Strain increment XX
EPSXX	NEL	Float	W	Strain XX
SIGOXX	NEL	Float	W	Old elasto-plastic stress XX
SIGNXX	NEL	Float	W	New elasto-plastic stress XX
SIGVXX	NEL	Float	W	Viscous stress XX
SOUNDSP	NEL	Float	W	Sound speed (time step)
VISCMAX	NEL	Float	W	Max damping modulus (time step)
THK	NEL	Float	R/W	Thickness
PLA	NEL	Float	R/W	Plastic strain
OFF	NEL	Float	R/W	Delete flag

Engine User's Arguments – Material Law for Bricks				
Variable	***Size***	***Type***	***Write/read***	***Definition***
NEL	1	Integer	R	Number of elements
UVAR	NUVAR* NEL	Float	R/W	User's variable
UPARAM	NUPARAM	Float	W	User's parameter
NUPARAM	1	Integer	W	Size of UPARAM
NUVAR	1	Integer	W	Number of user's variables
NFUNCT	1	Integer	W	Size of IFUNC
IFUNC	NFUNCT	Integer	W	Function number array
NPF	*	Integer	R	Function array
TF	*	Float	R	Function array
NGL	NEL	Integer	R	Element number
TIME	1	Float	R	Current time
TIMESTEP	1	Float	R	Current time step
RHO0	NEL	Float	R	Initial density
RHO	NEL	Float	R	Density
VOLUME	NEL	Float	R	Volume
EINT	NEL	Float	R	Total internal energy
EPSPXX	NEL	Float	W	Strain rate XX
DEPSXX	NEL	Float	W	Strain increment XX
EPSXX	NEL	Float	W	Strain XX
SIGOXX	NEL	Float	W	Old elasto-plastic stress XX
SIGNXX	NEL	Float	W	New elasto-plastic stress XX
SIGVXX	NEL	Float	W	Viscous stress XX
SOUNDSP	NEL	Float	W	Sound speed (time step)
VISCMAX	NEL	Float	W	Max damping modulus (time step)
OFF	NEL	Float	R/W	Delete flag

Engine User's Arguments – Spring Property				
Variable	***Size***	***Type***	***Write/read***	***Definition***
NEL	1	Integer	R	Number of elements
IPROP	1	Integer	R	Property number
UVAR	NUVAR* NEL	Float	R/W	User's variable
NUVAR	1	Integer	W	Number of user's variables
FX	NEL	Float	R/W	Tension force
FY	NEL	Float	R/W	Y shear force
FZ	NEL	Float	R/W	Z shear force
XMOM	NEL	Float	R/W	Torsion moment
YMOM	NEL	Float	R/W	Y bending moment
ZMOM	NEL	Float	R/W	Z bending moment
EINT	NEL	Float	R/W	Total internal energy
OFF	NEL	Float	R/W	Delete flag
STIFM	NEL	Float	W	Element stiffness (time step)
STIFR	NEL	Float	W	Element rotation stiffness (time step)

Engine User's Arguments – Spring Property				
Variable	***Size***	***Type***	***Write/read***	***Definition***
VISCM	NEL	Float	W	Element viscosity (time step)
VISCR	NEL	Float	W	Element rotation viscosity (time step)
MASS	NEL	Float	W	Element mass
XINER	NEL	Float	W	Element inertia (spherical)
DT	1	Integer	R	Time step
XL	NEL	Float	R	Element length
XL	NEL	Float	R	Element length
VX	NEL	Float	R	Tension velocity
RY1	NEL	Float	R	Node 1 Y bending rotational velocity
RZ1	NEL	Float	R	Node 1 Z bending rotational velocity
RX	NEL	Float	R	Torsional velocity
RY2	NEL	Float	R	Node 2 Y bending rotational velocity
RZ2	NEL	Float	R	Node 2 Z bending rotational velocity

Engine User's Arguments				
Variable	***Size***	***Type***	***Write/read***	***Definition***
NEL	1	Integer	R	Number of elements
IOUT	1	Integer	R	Output file unit (Engine output file)
UVAR	NUVAR* NEL	Float	R/W	User's variable
IPROP	1	Integer	R	Property number
IMAT	1	Integer	R	Material number
SOLID_ID	NEL	Integer	R	Solid Element ID
EINT	NEL	Float	R	Total internal energy
VOL	NEL	Float	R	Initial volume
OFF	NEL	Float	R/W	Delete flag
RHO	NEL	Float	R/W	Density
SIG	6* NEL	Float	R/W	Stress tensor
XX1	NEL	Float	R	X coordinate node 1
YY1	NEL	Float	R	Y coordinate node 1
ZZ1	NEL	Float	R	Z coordinate node 1
XX2 . . ZZ8	NEL	Float	R	Same for node 2 to 8
VX1	NEL	Float	R	X velocity node 1
VY1	NEL	Float	R	Y velocity node 1
VZ1	NEL	Float	R	Z velocity node 1
VRX1	NEL	Float	R	X rotational velocity node 1
VRY1	NEL	Float	R	Y rotational velocity node 1
VRZ1	NEL	Float	R	Z rotational velocity node 1
FX1	NEL	Float	W	X force node 1
FY1	NEL	Float	W	Y force node 1
FZ1	NEL	Float	W	Z force node 1
MX1	NEL	Float	W	X moment node 1
MY1	NEL	Float	W	Y moment node 1
MZ1	NEL	Float	W	Z moment node 1

Engine User's Arguments				
Variable	***Size***	***Type***	***Write/read***	***Definition***
STIFM	NEL	Float	W	Element stiffness (time step)
STIFR	NEL	Float	W	Element rotation stiffness (time step)
VISCM	NEL	Float	W	Element viscosity (time step)
VISCR	NEL	Float	W	Element rotation viscosity (time step)

Engine User's Arguments – Sensor				
Variable	***Size***	***Type***	***Write/read***	***Definition***
ID	NEL	Integer	R	Sensor ID

9.2 – Appendix 2 – Communication between User's Subroutines and RADIOSS Code

The main functions to access user's properties and materials are return functions and storage functions.

Return Function	Variables		Description
INTEGER *IP* = ***GET_U_P***(*PID*)	PID	Integer property ID	Returns a property number.
INTEGER *PID* = ***GET_U_PID***(*IP*)	IP	Property number	Returns the user property ID corresponding to the user property number IP.
FLOAT *PARAMI* = ***GET U GEO*** (*I,IP*)	I	Parameter index (1 for the first parameter, …)	Returns the user geometry parameters.
	IP	Property number	
INTEGER *II* = ***GET U PNU*** (*I,IP,KK*) *IFUNCI = GET_U_PNU (I,IP,KFUNC)* *IPROPI = GET_U_PNU(I,IP,KPROP)* *IMATI = GET_U_PNU(I,IP,KMAT)*	I	Parameter index	Returns the user-stored function (if KK=KFUNC), material (if KK=KMAT), or property (if KK=KPROP) numbers. See Starter subroutine for corresponding ID storage.
	IP	Property number	
	KK	Parameters KFUNC, KMAT, KPROP	
INTEGER *IM* = ***GET_U_M***(*MID*)	MID	Material ID	Returns a material number.
INTEGER *MID* = ***GET_U_MID***(*IM*)	IM	Material number	Returns the user material ID corresponding to user material number IM.
FLOAT *PARAMI* = ***GET_U_MAT***(*I,IM*)	I	Parameter index	Returns the user material parameters. **Note:** GET_U_MAT(0,IMAT) returns the density.
	IM	Material number	
INTEGER *IFUNCI* = ***GET_U_MNU*** *(I,IM,KFUNC)*	I	Variable index	Returns the user-stored function numbers (function referred by user's materials). See the Starter material user's subroutine for corresponding ID storage.
	IM	Material number	
	KFUNC	Only functions are yet available	
FLOAT *Y* = ***GET_U_FUNC***(*IFUNC,X,DYDX*)	IFUNC	Function numbered obtained by IFUNC = GET_U_MNU(I,IM, KFUNC) or IFUNC = GET_U_PNU(I,IP,KFUNC)	Returns Y(X).
	X	X value	
	DYDX	Slope dY/dX	

GET_TABLE_VALUE(NSENS, XX0,XDIM, tmp)

Storage Function	Description
INTEGER *ierror* = ***SET_U_PNU***(*func_index, fun_id, KFUNC)*	Stores RADIOSS function fun_id in the current user property buffer at a position referenced by func_index.
INTEGER *ierror* = ***SET_U_PNU***(*mat_index, mid, KMAT)*	Stores RADIOSS material mid in the current user property buffer at a position referenced by mat_index.
INTEGER *ierror* = ***SET_U_PNU***(*prop_index, pid, KPROP)*	Stores RADIOSS property pid in the current user property buffer at a position referenced by prop_index.
INTEGER *ierror* = ***SET_U_GEO***(*value_index, value)*	Stores a value in the current user property buffer at a position referenced by value_index.

Notes:

The geometry data (property) must be stored in RADIOSS storage with function SET_U_GEO(value_index,value).

If some standard RADIOSS functions (time or x,y) are used, function IDs must be stored with function SET_U_PNU(func_index,func_id,KFUNC).

If the property refers to user material, material IDs must be stored with function SET_U_PNU(mat_index,mat_id,KMAT).

If the property refers to user property, sub-property IDs must be stored with function SET_U_PNU(sub_prop_index,sub_prop_id,KPROP).

SET_U_GEO and SET_U_PNU return 0 if there are no errors.

SET_U_GEO and SET_U_PNU return the maximum allowed index if the index is larger than this maximum.

For TABLES use the subroutine: GET_TABLE_VALUE

INTEGER: TABLE_id = GET_NUMTABLE(TableID)

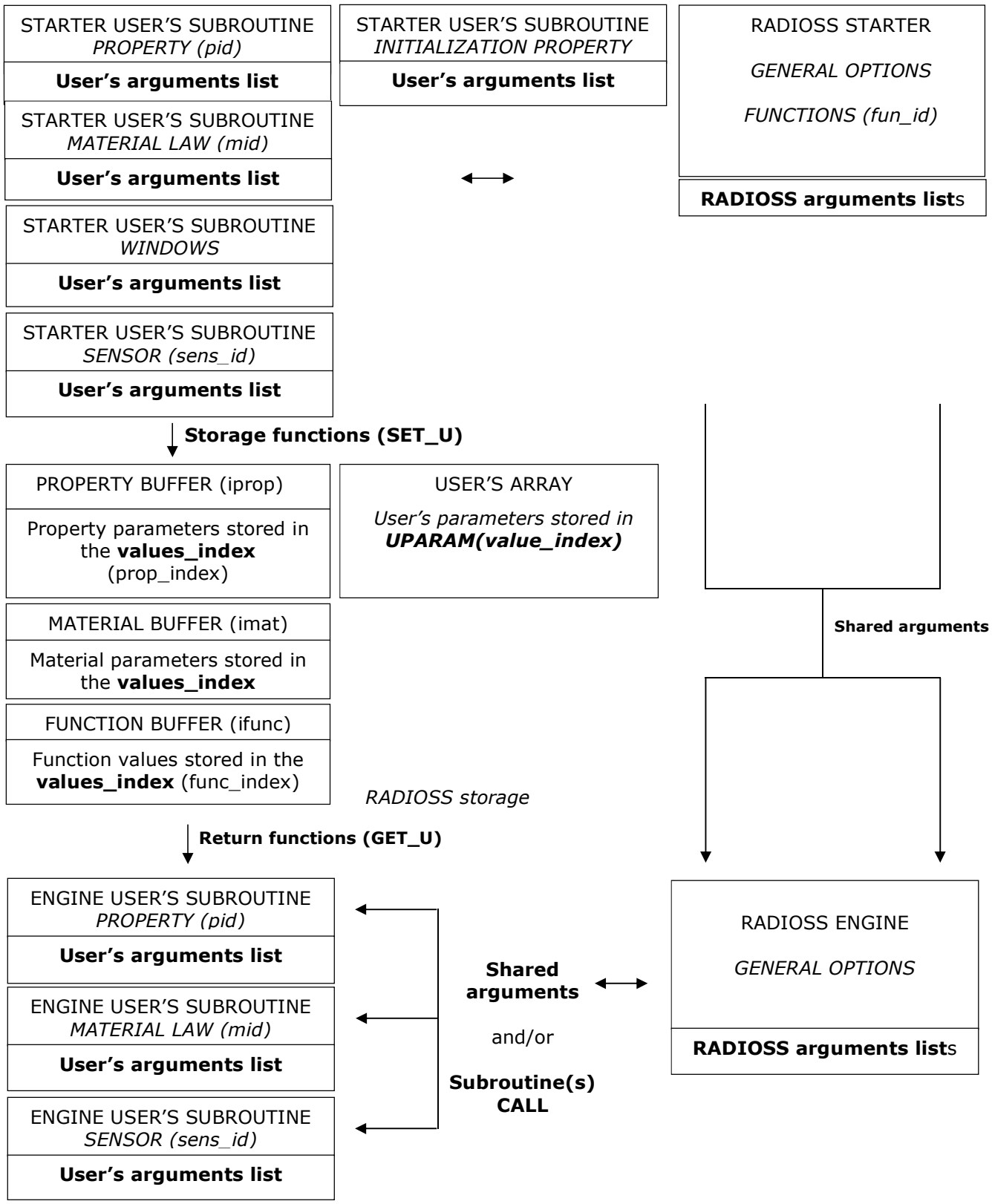
GET_TABLE_VALUE(TABLE_id, XX, X_DIM,YY), where XX is the input vector of dimension X_DIM and YY is the returned value.

Specific functions for user's subroutines type sensor include those below.

Function	Variables		Description
INTEGER NUM = ***GET_U_NUMSENS****(ID)*	ID	Sensor ID	Allows you to get a sensor number.
	NUM	Sensor number	
INTEGER *ierror =* ***GET_U_SENS_FPAR****(IVAR, VAR)*	IVAR	Float variable index	Allows you to retrieve a float sensor parameter.
	VAR	Integer value	
INTEGER ierror = ***GET_U_SENS_IPAR****(IVAR, VAR)*	IVAR	Integer variable index	Allows you to retrieve an integer sensor parameter.
	VAR	Integer value	
INTEGER *ierror =* ***SET_U_SENS_VALUE****(NSENS,IVAR,VAR)*	NSENS	Integer sensor number	Allows you to write a variable to a sensor buffer (float).
	IVAR	Integer buffer index	
	VAR	Float value	
INTEGER *ierror =* ***GET_U_SENS_VALUE****(NSENS,IVAR,VAR)*	NSENS	Integer sensor number	Allows you to read a variable from a sensor buffer (float).
	IVAR	Integer buffer index	
	VAR	Float value	
INTEGER ierror = ***SET_U_SENS_ACTI****(NSENS)*	NSENS	Integer sensor number	Allows you to activate a sensor using RADIOSS flag.
FLOAT *DTIME =* ***GET_U_SENS_ACTI****(NSENS)*	DTIME	Float time delay since first activation. (actif < = > DTIME > 0)	Allows you to check RADIOSS activation status of a sensor.
	NSENS	Integer sensor number	
FLOAT *TIME =* ***GET_U_TIME()***			Allows you to check the current time.
INTEGER *NCYC =* ***GET_U_CYCLE()***			Allows you to check the current cycle.
INTEGER *IERR =* ***GET_U_ACCEL****(NACC, AX,AY,AZ)*	AX, AY, AZ	Float acceleration components	Allows you to access accelerometer values.
	NACC	Integer accelerator number	
	IDACC	Integer accelerator ID	
	NACC	GET_U_NUMACC(IDACC)	
INTEGER *IERR =* *GET_U_NOD_X(NOD, X,Y,Z)* INTEGER *IERR =* *GET_U_NOD_D(NOD, DX,DY,DZ)* INTEGER *IERR =* *GET_U_NOD_V(NOD, VX,VY,VZ)*	X, Y, Z	Float nodal coordinates	Allows you access to nodal values.
	DX, DY, DZ	Float nodal displacements	
	VX, VY, VZ	Float nodal velocities	
	AX, AY, AZ	Float nodal accelerations	
	NOD	Integer node number	

Function	Variables		Description
INTEGER *IERR* = *GET_U_NOD_A(NOD, AX,AY,AZ)*	NID	Integer node ID	
	NOD	GET_U_NUMNOD(ID)	
FLOAT *Y* = ***GET_U_FUNC***(*IFUNC, X, DERI*)	X	Float ordinate	Allows you to access RADIOSS functions.
	Y	Float abscissa	
	DERI	Float Y/X	
	IFUNC	Integer function number	
	IDFUN	Integer function ID, where *INTEGER* IFUNC = GET_U_NUMFUN(IDFUN)	

Figure 2: Communication between user's modules and RADIOSS database



9.3 – Appendix 3 – General User's Subroutine Format with Material Laws 29, 30, and 31

9.3.1 Example Starter Format

```
      SUBROUTINE LECM29(IIN,IOUT,UPARAM,MAXUPARAM,NUPARAM,
     .                  NUVAR,IFUNC,MAXFUNC,NFUNC,PARMAT)
C
C-----------------------------------------------
C   D u m m y   A r g u m e n t s
C-----------------------------------------------
      INTEGER IIN,IOUT,MAXUPARAM,NUPARAM,NUVAR,MAXFUNC,NFUNC,
     .        IFUNC(MAXFUNC)
      DOUBLE PRECISION   UPARAM(MAXUPARAM),PARMAT(*)
C
C-----------------------------------------------
C   L o c a l   V a r i a b l e s
C-----------------------------------------------
      DOUBLE PRECISION E,NU,G,A11
C
C-----------------------------------------------
C   U s e r ' s   D a t a   R e a d i n g
C-----------------------------------------------
      READ(IIN,'(2F20.0)')E,NU
      A11 = E * (1.-NU) / (1.+NU) / (1.-2.*NU)
      A12 = E * NU / (1.+NU) / (1.-2.*NU)
C
C-----------------------------------------------
C   C h e c k
C-----------------------------------------------
      IF(NU.LT.0.0.OR.NU.GE.0.5)THEN
        WRITE(IOUT,*)' ** ERROR : WRONG NU VALUE'
      ENDIF
C
C-----------------------------------------------
C   U s e r ' s   A r g u m e n e n t s   D e f i n i t i o n
C-----------------------------------------------
       NUVAR = 2
       NFUNC = 0
       NUPARAM = 3
       UPARAM(1) = E
       UPARAM(2) = A11
       UPARAM(3) = A12
C
C-----------------------------------------------
C   O u t p u t   F i l e   P r i n t
C-----------------------------------------------
      WRITE(IOUT,1000)
      WRITE(IOUT,1100)E,NU
1000 FORMAT(
     & 5X,'  MATERIAL LAW 29',/,
1100 FORMAT(
     & 5X,'E . . . . . . . . . . . . . . . .=',E12.4/
     & 5X,'NU. . . . . . . . . . . . . . . .=',E12.4//)
C
      RETURN
      END
```

Standardized headers:
User's arguments list
(read, written or read/written)

Arguments:
Used in communications between RADIOSS database and user's subroutines

Local variables:
Used in current user's subroutine

User's data reading:
Reads input user's data specified in RADIOSS Starter input file (user's options: */MAT/USER1/)*

Check:
Write message

Argument definition:
Stores user's arguments values as user's parameters

There can be returned by a restore function in the Engine user's subroutine

File print:
Prints user's values in Starter output file

9.3.2 Example Engine Format

```
      SUBROUTINE SIGEPS29C(
     1    NEL    ,NUPARAM,NUVAR   ,NFUNC   ,IFUNC  ,
     2    NPF    ,NPT    ,IPT     ,IFLAG   ,
     2    TF     ,TIME   ,TIMESTEP,UPARAM  ,RHO0   ,
     3    AREA   ,EINT   ,THKLY   ,
     4    EPSPXX ,EPSPYY ,EPSPXY  ,EPSPYZ  ,EPSPZX ,
     5    DEPSXX ,DEPSYY ,DEPSXY  ,DEPSYZ  ,DEPSZX ,
     6    EPSXX  ,EPSYY  ,EPSXY   ,EPSYZ   ,EPSZX  ,
     7    SIGOXX ,SIGOYY ,SIGOXY  ,SIGOYZ  ,SIGOZX ,
     8    SIGNXX ,SIGNYY ,SIGNXY  ,SIGNYZ  ,SIGNZX ,
     9    SIGVXX ,SIGVYY ,SIGVXY  ,SIGVYZ  ,SIGVZX ,
     A    SOUNDSP,VISCMAX,THK     ,PLA     ,UVAR   ,
     B    OFF    ,NGL    ,SHF)
C
C-------------------------------------------------
C   I N P U T   A r g u m e n t s
C-------------------------------------------------
      INTEGER NEL,NUPARAM,NUVAR
      DOUBLE PRECISION  UPARAM(NUPARAM),
     .   DEPSXX(NEL),DEPSYY(NEL),
     .   SIGOXX(NEL),SIGOYY(NEL)
C
C-------------------------------------------------
C   O U T P U T   A r g u m e n t s
C-------------------------------------------------
      DOUBLE PRECISION SIGNXX(NEL),SIGNYY(NEL)
C
C-------------------------------------------------
C   I N P U T   O U T P U T   A r g u m e n t s
C-------------------------------------------------
      DOUBLE PRECISION UVAR(NEL,NUVAR)
C
C-------------------------------------------------
C   L o c a l   V a r i a b l e s
C-------------------------------------------------
      INTEGER I
      DOUBLE PRECISION A1,A2
C
C-------------------------------------------------
C   U s e r ' s   P r o g r a m
C-------------------------------------------------
      DO I=1,NEL
        A1  = UPARAM(2)
        A2  = UPARAM(3)
        SIGNXX(I)=SIGOXX(I)+A1*DEPSXX(I)+A2*DEPSYY(I)
        SIGNYY(I)=SIGOYY(I)+A2*DEPSXX(I)+A1*DEPSYY(I)
        UVAR(I,1) = SIGNXX(I)
        UVAR(I,2) = SIGNYY(I)
      ENDDO
C
      RETURN
      END
```

Standardized headers:
User's arguments list
(read, written or read/written)

Input arguments:
Used in current routine (user's program)

Output arguments:
Computes in current routine

User's variable:
Used in communications between user's subroutines

Local variables:
Used in current user's subroutine

User's program (on elements: NEL):
Reads input arguments and writes output arguments

The user's variables UVAR can be saved in animations file:
UVAR(I,1) saved as User Var 1 in animations using */ANIM/ELEM/USER*. option in Engine input file.

9.4 – Appendix 4 – General User's Subroutine Format with Extended User Material Laws

9.4.1 Example Starter Format

```
SUBROUTINE LECMUSERnn(IIN IOUT,UPARAM,MAXUPARAM,NUPARAM,
    .NUVAR,IFUNC,MAXFUNC,NFUNC,PARMAT)
C
C-------------------------------------------
C   D u m m y   A r g u m e n t s
C-------------------------------------------
      INTEGER IIN,IOUT,MAXUPARAM,NUPARAM,NUVAR,MAXFUNC,NFUNC,
     .        IFUNC(MAXFUNC)
      DOUBLE PRECISION   UPARAM(MAXUPARAM),PARMAT(*)
C
C-------------------------------------------
C   L o c a l   V a r i a b l e s
C-------------------------------------------
      DOUBLE PRECISION E,NU,G,A11
C
C-------------------------------------------
C   U s e r ' s   D a t a   R e a d i n g
C-------------------------------------------
      READ(IIN,'(2F20.0)')E,NU
      A11 = E * (1.-NU) / (1.+NU) / (1.-2.*NU)
      A12 = E * NU / (1.+NU) / (1.-2.*NU)
C
C-------------------------------------------
C   C h e c k
C-------------------------------------------
      IF(NU.LT.0.0.OR.NU.GE.0.5)THEN
        WRITE(IOUT,*)' ** ERROR : WRONG NU VALUE'
      ENDIF
C
C-------------------------------------------
C   U s e r ' s   A r g u m e n e n t s   D e f i n i t i o n
C-------------------------------------------
       NUVAR = 2
       NFUNC = 0
       NUPARAM = 3
       UPARAM(1) = E
       UPARAM(2) = A11
       UPARAM(3) = A12
C
C-----------------------------------------------
C   O u t p u t   F i l e   P r i n t
C-----------------------------------------------
      WRITE(IOUT,1000)
      WRITE(IOUT,1100)E,NU
1000 FORMAT(
     & 5X,'  MATERIAL LAW 12',/)
1100 FORMAT(
     & 5X,'E . . . . . . . . . . . . . .=',E12.4/
     & 5X,'NU. . . . . . . . . . . . . .=',E12.4//)C
      RETURN
      END
```

Standardized headers:
User's arguments list
(read, written or read/written)

Arguments:
Used in communications between RADIOSS database and user's subroutines

Local variables:
Used in current user's subroutine

User's data reading:
Reads input user's data specified in RADIOSS Starter input file
(user's options: */MAT/USER1/)*

Check:
Write message

Argument definition:
Stores user's arguments values as user's parameters

There can be returned by a restore function in the Engine user's subroutine

File print:
Prints user's values in Starter output file

9.4.2 Example Engine Format

```
      SUBROUTINE LUSERnnC(
     1     NEL    ,NUPARAM,NUVAR   ,NFUNC   ,IFUNC  ,
     2     NPF    ,NGL
     2     TF     ,TIME    ,TIMESTEP,UPARAM  ,RHO0   ,
     3     AREA   ,EINT    ,SHF     ,
     4     SOUNDSP,VISCMAX     ,PLA     ,UVAR   ,
     5     OFF    ,SIGY    ,USERBUF)
C
C-----------------------------------------------
C   I N P U T   A r g u m e n t s
C-----------------------------------------------
      INTEGER NEL,NUPARAM,NUVAR
      DOUBLE PRECISION  UPARAM(NUPARAM),
     .   DEPSXX(NEL),DEPSYY(NEL),
     .   SIGOXX(NEL),SIGOYY(NEL)
C
C-----------------------------------------------
C   O U T P U T   A r g u m e n t s
C-----------------------------------------------
      DOUBLE PRECISION SIGNXX(NEL),SIGNYY(NEL)
C
C-----------------------------------------------
C   I N P U T   O U T P U T   A r g u m e n t s
C-----------------------------------------------
      DOUBLE PRECISION UVAR(NEL,NUVAR)
C
C-----------------------------------------------
C   L o c a l   V a r i a b l e s
C-----------------------------------------------
      INTEGER I
      DOUBLE PRECISION A1,A2
C
C-----------------------------------------------
C   U s e r ' s   P r o g r a m
C-----------------------------------------------
      DO I=1,NEL
        A1  = UPARAM(2)
        A2  = UPARAM(3)
        SIGNXX(I)=SIGOXX(I)+A1*DEPSXX(I)+A2*DEPSYY(I)
        SIGNYY(I)=SIGOYY(I)+A2*DEPSXX(I)+A1*DEPSYY(I)
        UVAR(I,1) = SIGNXX(I)
        UVAR(I,2) = SIGNYY(I)
      ENDDO
C
      RETURN
      END
```

Standardized headers:
User's arguments list
(read, written or read/written)

Input arguments:
Used in current routine (user's program)

Output arguments:
Computes in current routine

User's variable:
Used in communications between user's subroutines.

Local variables:
Used in current user's subroutine

User's program (on elements: NEL):
Reads input arguments and writes output arguments

The user's variables UVAR can be saved in animations file:
UVAR(I,1) saved as User Var 1 in animations using */ANIM/ELEM/USER1* option in 0001.rad.